Object Oriented Programming

Lecture No. 5

Ing. Lukáš Slánský

Univerzita Pardubice

24. 11. 2008

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Contents

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Semestrální práce

- Zadání byla zveřejněna RNDr. Rakem na STAGu (portálu) – zadání jsou záměrně vágní.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Semestrální práce

- Zadání byla zveřejněna RNDr. Rakem na STAGu (portálu) – zadání jsou záměrně vágní.
- Skupiny maximálně o velikosti 4 studenti – nahlásit emailem společně s vybraným tématem do 30.11.2008.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Semestrální práce

- Zadání byla zveřejněna RNDr. Rakem na STAGu (portálu) – zadání jsou záměrně vágní.
- Skupiny maximálně o velikosti 4 studenti – nahlásit emailem společně s vybraným tématem do 30.11.2008.
- Rozdělení do skupin, hodnocení, výsledky atd. jsou přístupné na adrese `http://spreadsheets.google.com/pub?key=p8w2JsU58MBoBSl1cTu5WTg`.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Požadavky na semestrální práci

- Program

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Požadavky na semestrální práci

- Program
  - Kvalitní objektová struktura
  - Prezistence dat, je-li smysluplná
  - Vícevrstvá architektura, je-li smysluplná

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Požadavky na semestrální práci

- Program
  - Kvalitní objektová struktura
  - Prezistence dat, je-li smysluplná
  - Vícevrstvá architektura, je-li smysluplná
- Programátorská dokumentace
  - Přehled struktury programu – za pomoci UML diagramů (min. třídní diagramy)
  - Popis významu (ne fungování!) jednotlivých bloků (knihoven, tříd, atributů, metod, proměnných, . . .)

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Požadavky na semestrální práci

- Program
  - Kvalitní objektová struktura
  - Prezistence dat, je-li smysluplná
  - Vícevrstvá architektura, je-li smysluplná
- Programátorská dokumentace
  - Přehled struktury programu – za pomoci UML diagramů (min. třídní diagramy)
  - Popis významu (ne fungování!) jednotlivých bloků (knihoven, tříd, atributů, metod, proměnných, . . .)
- Uživatelská dokumentace
  - Příručka pro koncové uživatele
  - Přehled ovládání programu, omezení, chybových hlášení, . . .

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.
   - *Interface*

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.
   - *Interface* – public set of messages.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.
   - *Interface* – public set of messages.
   - *Protocol*

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.
   - *Interface* – public set of messages.
   - *Protocol* – mapping (unique assignment) of interface to set of methods.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.
   - *Interface* – public set of messages.
   - *Protocol* – mapping (unique assignment) of interface to set of methods.
4. Object can *c*ontain other objects.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Axiomatic Definition of Objects

Definition: Object is an independent program structure defined acording to axioms:

1. Object contains *attributes* (variables)
   - = private (not public) memory of the object where is stored actual state of the object.
2. Object contains *methods* (procedures and functions)
   - = private (not public) operations with attributes that change state of the object.
3. Object is capable of *receiving and process message*.
   - *Interface* – public set of messages.
   - *Protocol* – mapping (unique assignment) of interface to set of methods.
4. Object can *c*ontain other objects.
   - You can send messages to other objects. Messages control other objects.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

UML

- *UML* (Unified Modelling Language) is a "language" that is used to specify, visual description, documentation and partial implementation of SW systems parts.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

UML

- *UML* (Unified Modelling Language) is a "language" that is used to specify, visual description, documentation and partial implementation of SW systems parts.
- It has been founded in 1990s as a standard of modelling language. The specification has been accepted by OMG (Object Management Group) that associates companies like DEC, HP, MS, Oracle, IBM, and many others.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

UML

- *UML* (Unified Modelling Language) is a "language" that is used to specify, visual description, documentation and partial implementation of SW systems parts.
- It has been founded in 1990s as a standard of modelling language. The specification has been accepted by OMG (Object Management Group) that associates companies like DEC, HP, MS, Oracle, IBM, and many others.
- It is used to systems modelling from overall overview till details like attributes of particular classes.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Basic Properties of UML

- It is easy to use visual modelling language that allows developers to develop and interchange models.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Basic Properties of UML

- It is easy to use visual modelling language that allows developers to develop and interchange models.
  - It must be simple but very powerfull.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Basic Properties of UML

- It is easy to use visual modelling language that allows developers to develop and interchange models.
  - It must be simple but very powerfull.
- It is extensible and implements specialisation that allows extension of basic concepts.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Basic Properties of UML

- It is easy to use visual modelling language that allows developers to develop and interchange models.
  - It must be simple but very powerfull.
- It is extensible and implements specialisation that allows extension of basic concepts.
- It is indenpendent of the programing languages.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Basic Properties of UML

- It is easy to use visual modelling language that allows developers to develop and interchange models.
  - It must be simple but very powerfull.
- It is extensible and implements specialisation that allows extension of basic concepts.
- It is indenpendent of the programing languages.
- It supports higher level concepts – cooperations, templates, ...

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

UML Diagram Overview

- Structure Diagrams:
  - Class Diagram,
  - Object Diagram,
  - Component Diagram,
  - Deployment Diagram,
  - Package Diagram,
  - Composite Structure Diagram.
- Behaviour Diagrams:
  - Use-case Diagram,
  - State Machine Diagram,
  - Sequence Diagram,
  - Activity Diagram,
  - Communication Diagram,
  - Timing Diagram (in UML 2.0),
  - Interaction Overview Diagram (in UML 2.0).

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Class Diagram

Class Diagram shows particular classes in system:

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Class Diagram

Class Diagram shows particular classes in system:

- Attributes, methods:
  - parameters, return types,
  - access restrictions.
- Relations between classes:

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Class Diagram

Class Diagram shows particular classes in system:

- Attributes, methods:
  - parameters, return types,
  - access restrictions.
- Relations between classes:
  - inheritance,
  - association, aggregation, composition.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Classes and Objects in UML

- Class is written as rectangle.

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Classes and Objects in UML

- Class is written as rectangle.
  - Class name is written in bold with capital on first leter in all words of its' name.

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Classes and Objects in UML

- Class is written as rectangle.
  - Class name is written in bold with capital on first leter in all words of its' name.
- Object is written as rectangle too.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Classes and Objects in UML

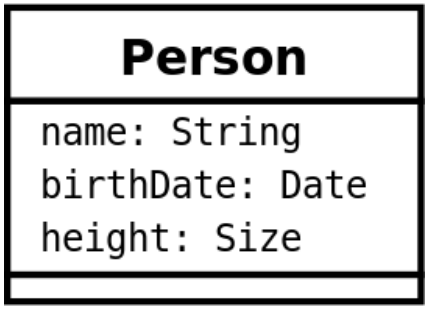
- Class is written as rectangle.
  - Class name is written in bold with capital on first leter in all words of its' name.
- Object is written as rectangle too.
  - Object name is in regular and underlined form with small letter in the beginning,
  - class name is behind objects' name (like in C).

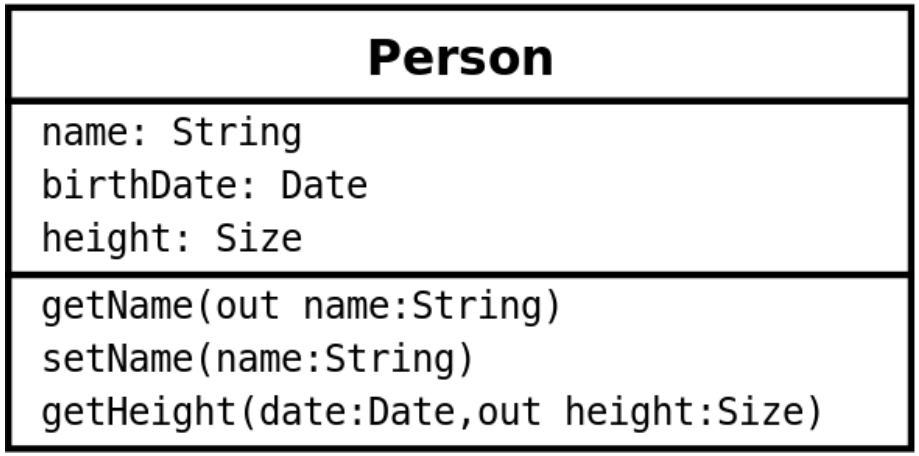
Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Classes and Objects in UML

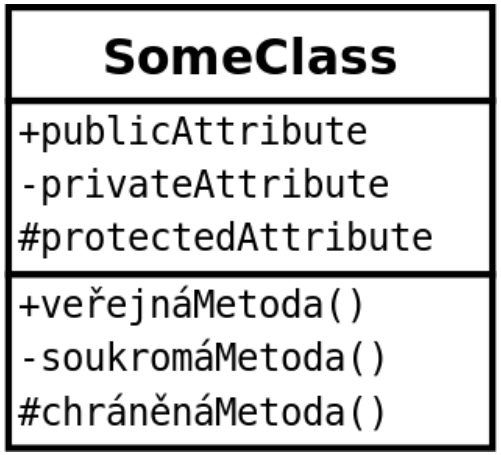
- Class is written as rectangle.
  - Class name is written in bold with capital on first leter in all words of its' name.
- Object is written as rectangle too.
  - Object name is in regular and underlined form with small letter in the beginning,
  - class name is behind objects' name (like in C).

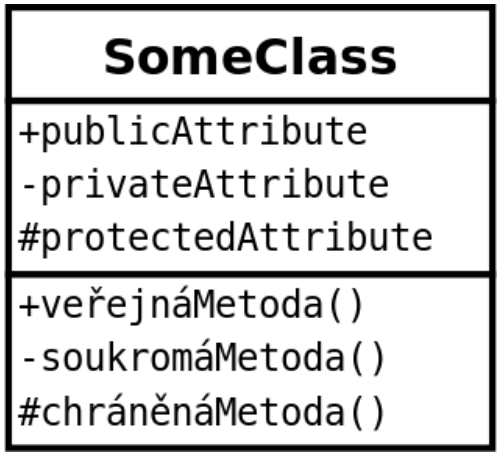
Class in UML

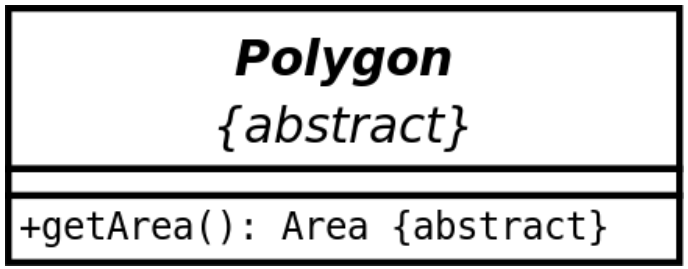




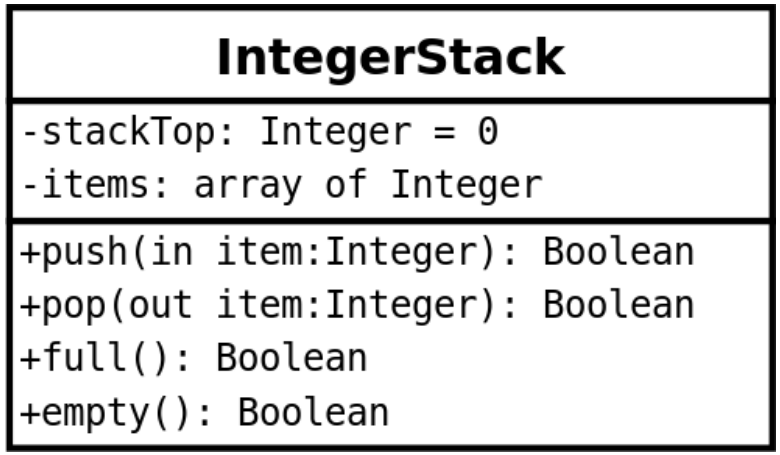
Univerzita Pardubice
Fakulta elektrotechniky a informatiky


SomeClass
+publicAttribute
-privateAttribute
#protectedAttribute
+veřejnáMetoda()
-soukromáMetoda()
#chráněnáMetoda()

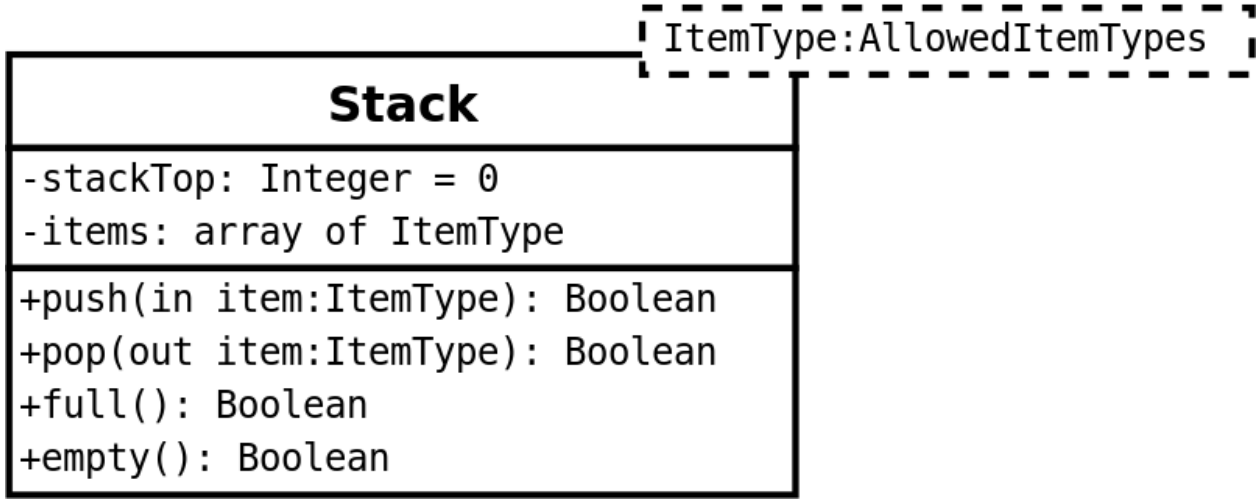

SomeClass
+publicAttribute
-privateAttribute
#protectedAttribute
+veřejnáMetoda()
-soukromáMetoda()
#chráněnáMetoda()


Polygon
{abstract}
+getArea(): Area {abstract}

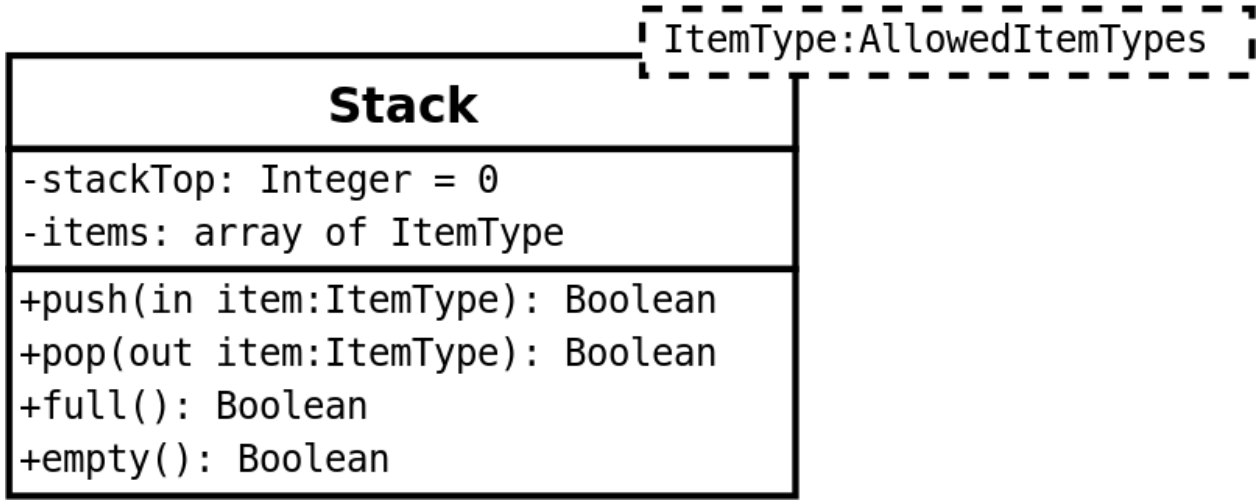
Stack in UML – I.



Stack in UML – II.



Stack in UML – II.



Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Tools for System Design in UML

- Commercial products
  - IBM Rational Rose (`http://www-306.ibm.com/software/rational/`) – price 200000-350000 CZK
  - Sybase PowerDesigner (`http://www.sybase.com/products/developmentintegration/powerdesigner`) – price 270000 CZK
  - Many further product made by Borland, ARTiSAN, ...

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Tools for System Design in UML

- Commercial products
  - IBM Rational Rose (`http://www-306.ibm.com/software/rational/`) – price 200000-350000 CZK
  - Sybase PowerDesigner (`http://www.sybase.com/products/developmentintegration/powerdesigner`) – price 270000 CZK
  - Many further product made by Borland, ARTiSAN, ...
- Product with GPL or similar license
  - ArgoUML (`http://argouml.tigris.org/`)
    - ★ Commercial branch is Poseidon for UML (`http://gentleware.com/index.php`)
  - Umbrello (`http://uml.sourceforge.net/`)
  - Violet (`http://www.horstmann.com/violet/`)

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Student Lectures

- Last lecture will be dedicated to lectures of some themes with some relation to OOP.

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Student Lectures

- Last lecture will be dedicated to lectures of some themes with some relation to OOP.
- Length cca 15-20 minutes for 1 or 2 students.

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Student Lectures

- Last lecture will be dedicated to lectures of some themes with some relation to OOP.
- Length cca 15-20 minutes for 1 or 2 students.
- 10 points bonus for exam.

Univerzita
Pardubice
Fakulta elektrotechniky
a informatiky

Student Lectures

- Last lecture will be dedicated to lectures of some themes with some relation to OOP.
- Length cca 15-20 minutes for 1 or 2 students.
- 10 points bonus for exam.
- Themes:
  1. Design Patterns,
  2. CORBA, DCOM, Java-RMI – concept and comparison,
  3. Modern System Developement (Component Developement, Aspect-oriented Programming, . . .).

Univerzita Pardubice
Fakulta elektrotechniky a informatiky